M16C/60 Series PC7501 Emulator

C0 Coverage Facility

Summary

The full-spec emulator PC7501 for the M16C/60 series incorporates the C0 coverage facility. This document explains how to use the C0 coverage facility while using the PC7501 emulator.

Explained in this document is for the case where the user system incorporating an M16C/60 series microcomputer and the PC7501 emulator are used in combination. The content of this document may be used in common for even a different target microcomputer providing it is one of the M16C/60 series microcomputers.

Table of Contents

1. Overview 2

2. Facilities Used 2

3. Preparing the Software 2
3.1 Getting Started 2
3.2 Installing the Software Supplied with the PC7501 Emulator 2
3.3 Auto Update Utility 2

4. Operational Description 3
4.1 Starting the High-performance Embedded Workshop 3
4.2 Opening a Workspace 4
4.3 Downloading the Target Program 8
4.4 Measuring the C0 Coverage 9

5. Frequently Asked Questions 18
5.1 Why do the measurement result in source units and that in the coverage window differ? 18

6. Related Documents 20

1. Overview

The coverage facility refers to the one that measures the extent of a test covered. The coverage rate of a test program measured while it was run is used to determine the validity of the test program. The coverage facility comes in several types such as C0 coverage and C1 coverage. Incorporated in the PC7501 is the C0 coverage. Also known as instruction coverage, the C0 coverage allows measuring the coverage of whether instructions were executed. On the other hand, the C1 coverage, also referred to as branch coverage, allows measuring the coverage of whether a branch occurred on true or false side of a conditional branch instruction.

The PC7501 can not only display the result of a C0 coverage measurement in the C0 coverage window, but also permits you to verify instruction execution or non-execution in source line units using the editor window columns, as well as verify instruction execution or non-execution in address units in the memory window. The coverage measurement area can be any area comprised of 1 to 32 blocks (up to 8 Mbytes) beginning with the 256-kbyte boundary. There is another thing to be noted for the PC7501's C0 coverage facility that its measurement result includes prefetches.

2. Facilities Used

In this document, the method of C0 coverage measurement is explained using the sample program included in the CD-ROM supplied with the PC7501 emulator or a downloaded package from the Renesas website.

The tool versions used are listed below.

M16C R8C PC7501 Emulator Debugger V.1.03 Release 00 for the M16C series

3. Preparing the Software

3.1 Getting Started

Install the software included in the CD-ROM that is supplied with the PC7501 emulator.

The sample program (tutorial workspace) used in this document will be expanded in your PC.

The software included in the CD-ROM that is supplied with the PC7501 emulator may also be installed in a PC that has had the High-performance Embedded Workshop already installed. In this case, part of the dialogs displayed during the installation work will be omitted.

3.2 Installing the Software Supplied with the PC7501 Emulator

Execute M16cPc7501 Debugger.exe present in the CD-ROM supplied with the PC7501 emulator.

For details on how to install, see the PC7501 emulator setup guide published at the Renesas website. During installation work, follow the instructions displayed on the screen. The installation procedure is omitted here.

3.3 Auto Update Utility

If you've selected the auto update utility when installing the software, it is possible to confirm via the Internet whether the latest version of each tool is available.

4. Operational Description

This section describes the method for using the C0 coverage facility after starting the High-performance Embedded Workshop (HEW). The procedure is shown below.

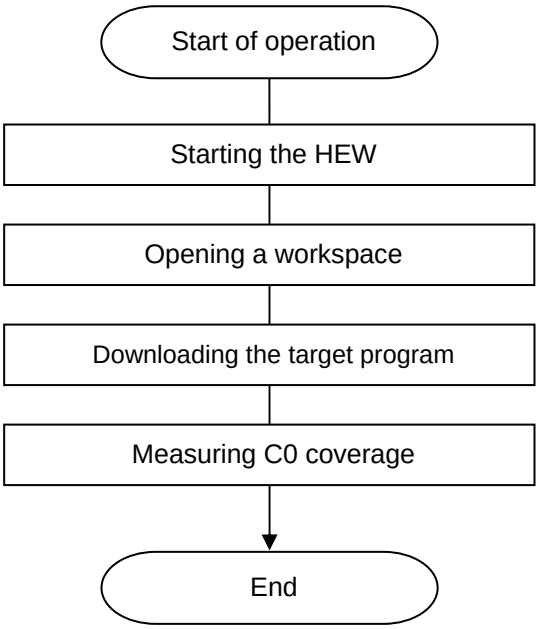


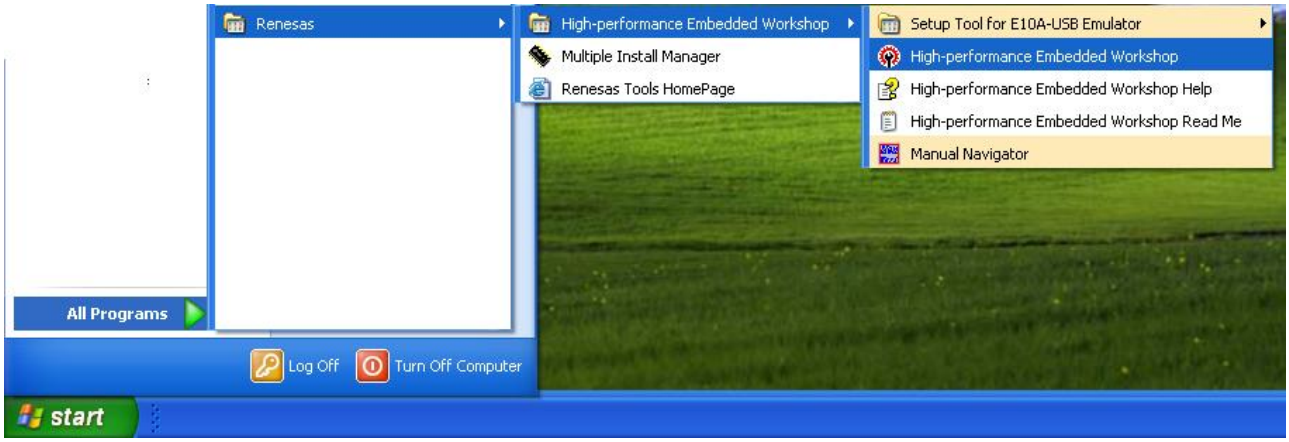
Figure 4.1 Example Program Execution Procedure

4.1 Starting the High-performance Embedded Workshop

To begin with, first connect the PC7501 emulator that has the user system connected to it and the host computer with USB cable, and check to see that everything is ready to debug.

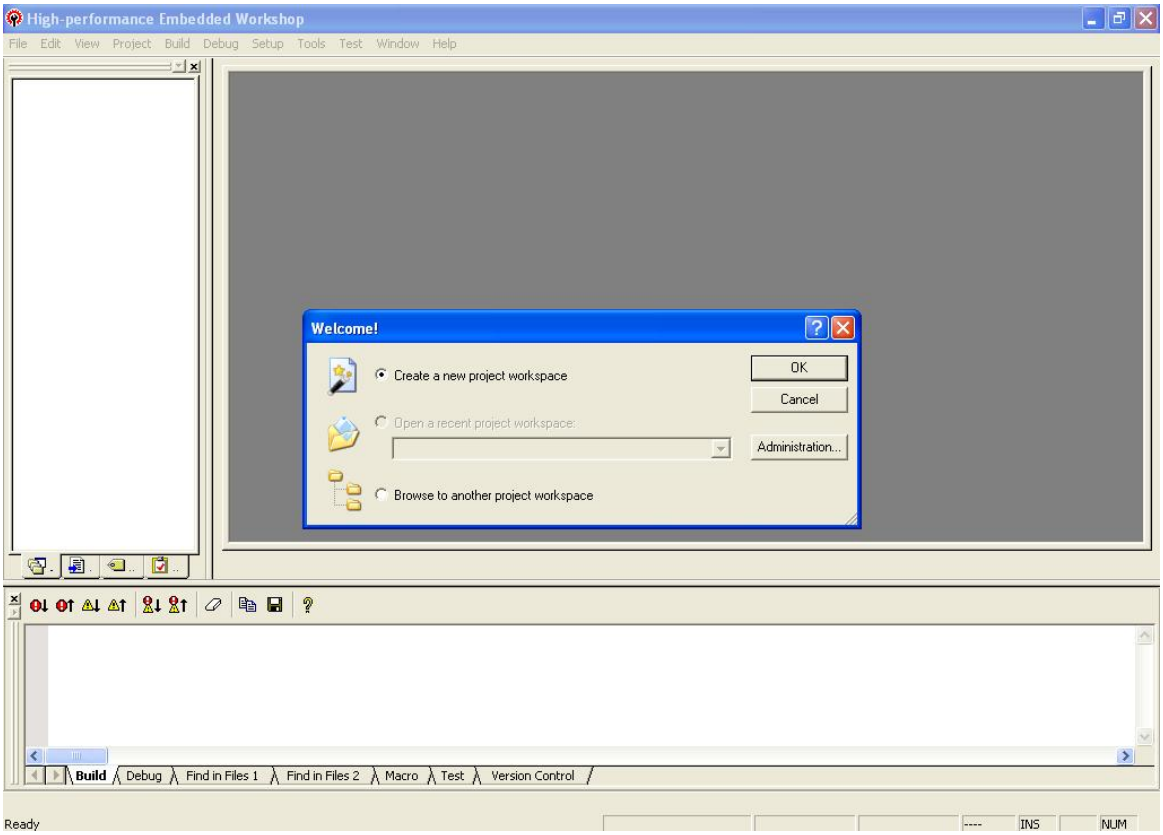
Next, start the High-performance Embedded Workshop.

From All Programs on the Start menu, choose Renesas → High-performance Embedded Workshop → High-performance Embedded Workshop, to start.

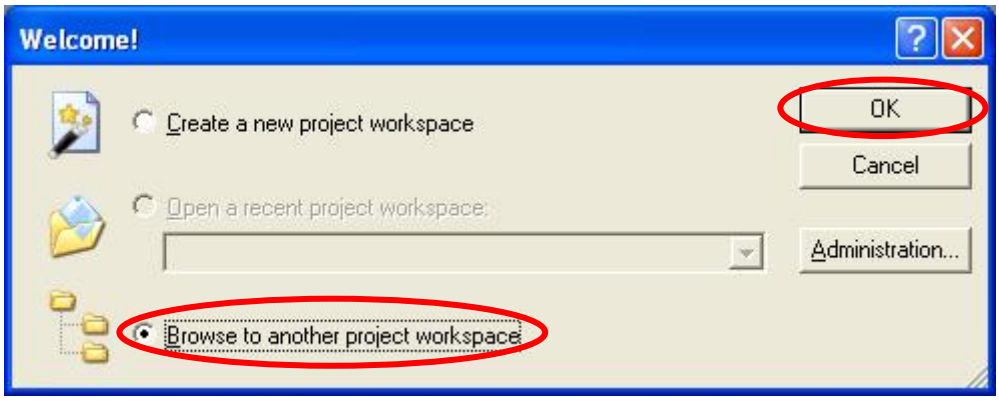


4.2 Opening a Workspace

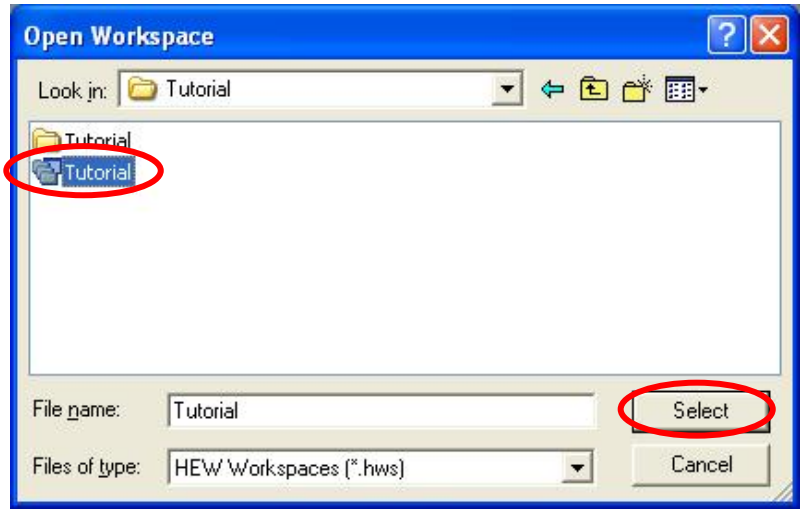
(1) The Welcome! dialog box will be displayed in the High-performance Embedded Workshop.



Select the “Browse Another Project Workspace” radio button in the Welcome! dialog box and click the OK button.



(2) The Open a Workspace dialog box will be displayed.



If the installation of the CD-ROM of this product is complete, the workspace “Tutorial.hws” is stored as standard in the folder position shown below. Check folder positions in order while you locate. When the workspace “Tutorial.hws” is found, specify it and click the Select button.

C:\WorkSpace\Tutorial\PC7501\M16C\Tutorial\Tutorial.hws
C:\WorkSpace └Tutorial └PC7501 └M16C └Tutorial └Tutorial.hws

[Note] Depending on the software version used, it will occur that the above directory cannot be specified. In such a case, select the directory given below.

<Directory in which the High-performance Embedded Workshop is installed>

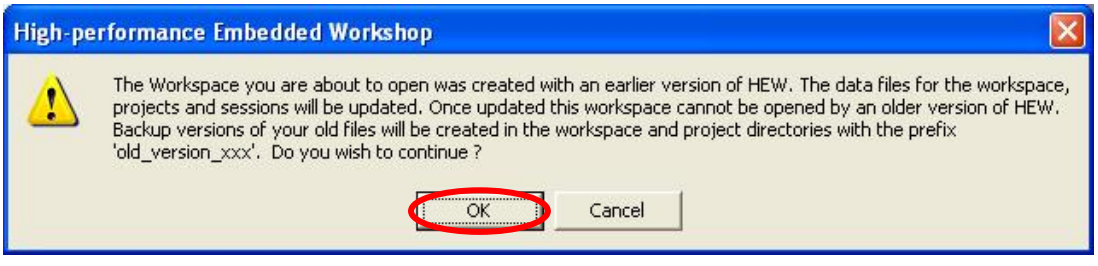
\Tools\Renesas\DebugComp\Platform\E8\M16C\Tutorial

Examples of directory:

C:\hew3\Tools\Renesas\DebugComp\Platform\E8\M16C\Tutorial

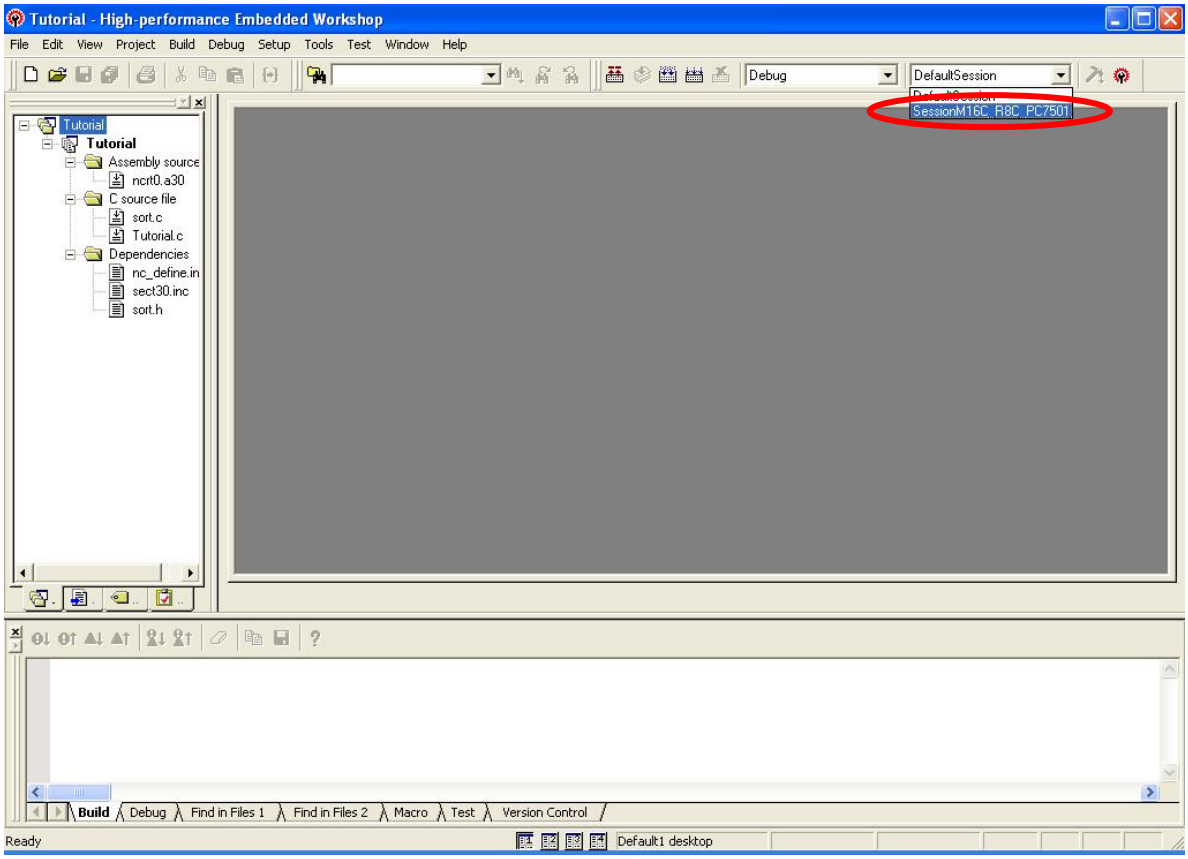
C:\hew2\Tools\Renesas\DebugComp\Platform\E8\M16C\Tutorial

(3) If the workspace version is old, the dialog box shown below is displayed. To update it to a new version, click the OK button.

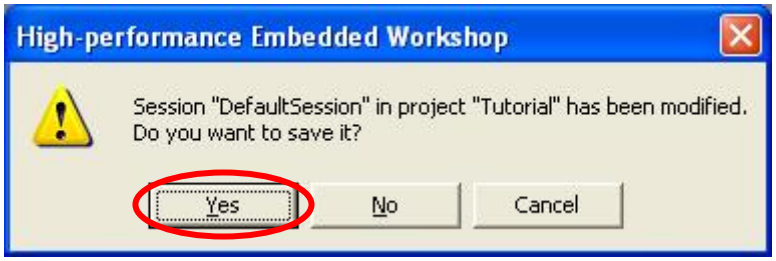

High-performance Embedded Workshop
The Workspace you are about to open was created with an earlier version of HEW. The data files for the workspace, projects and sessions will be updated. Once updated this workspace cannot be opened by an older version of HEW. Backup versions of your old files will be created in the workspace and project directories with the prefix 'old_version_xxx'. Do you wish to continue ?
OK
Cancel

(4) When the workspace is opened, you are ready to use the High-performance Embedded Workshop. Change sessions to connect the PC7501 emulator. To do it, change “DefaultSession” in the toolbar to “SessionM16C_R8C_PC7501_Emulator.”

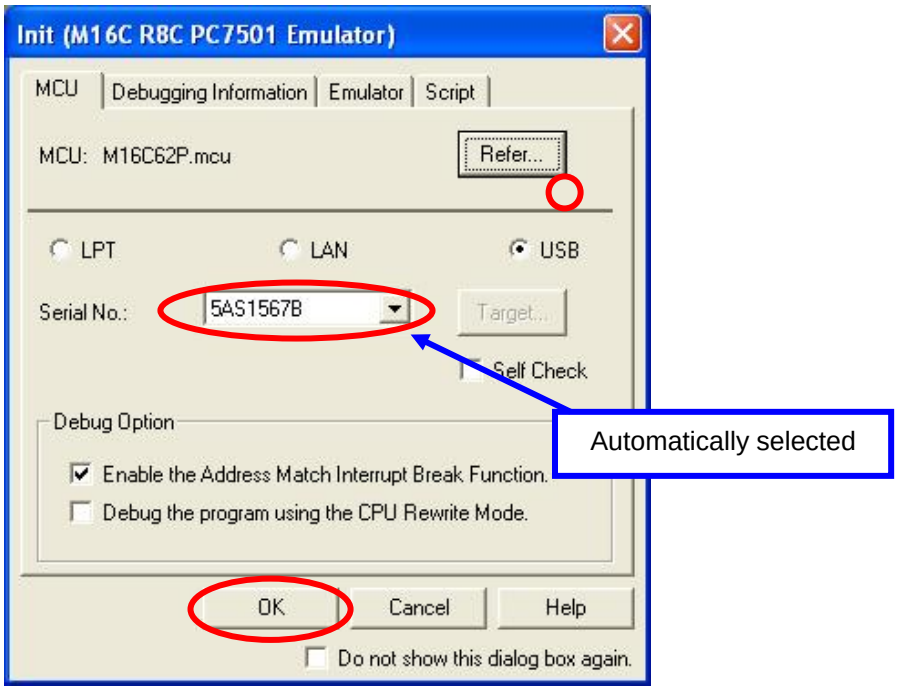
Before changing sessions here, be sure that the interface select switch on the back panel of the PC7501 emulator main body is set to the USB side, and then turn the power for the PC7501 emulator on.



If a dialog box is displayed prompting for your confirmation of whether to save the previous session, click the Yes button.

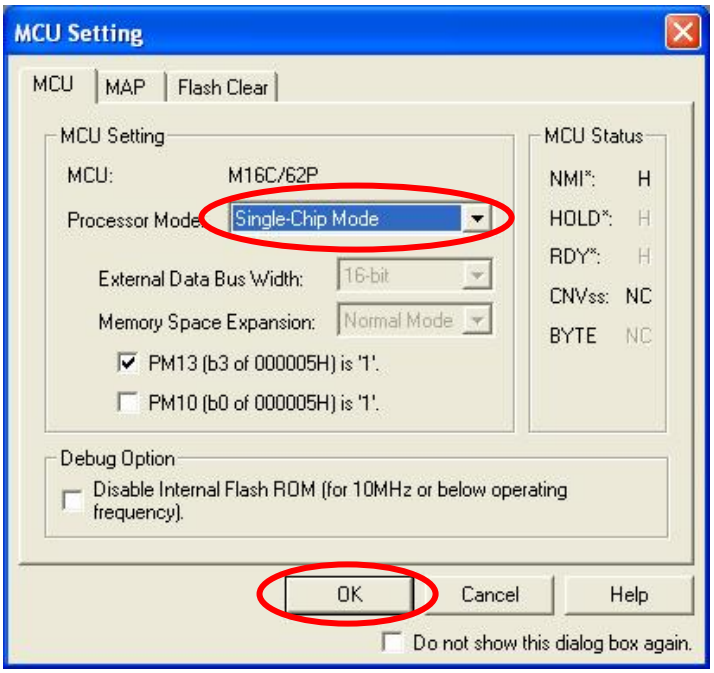


(5) The Init (M16C R8C PC7501 Emulator) dialog box will be displayed. Select the “USB” radio button on the MCU tab of the dialog box and click OK.

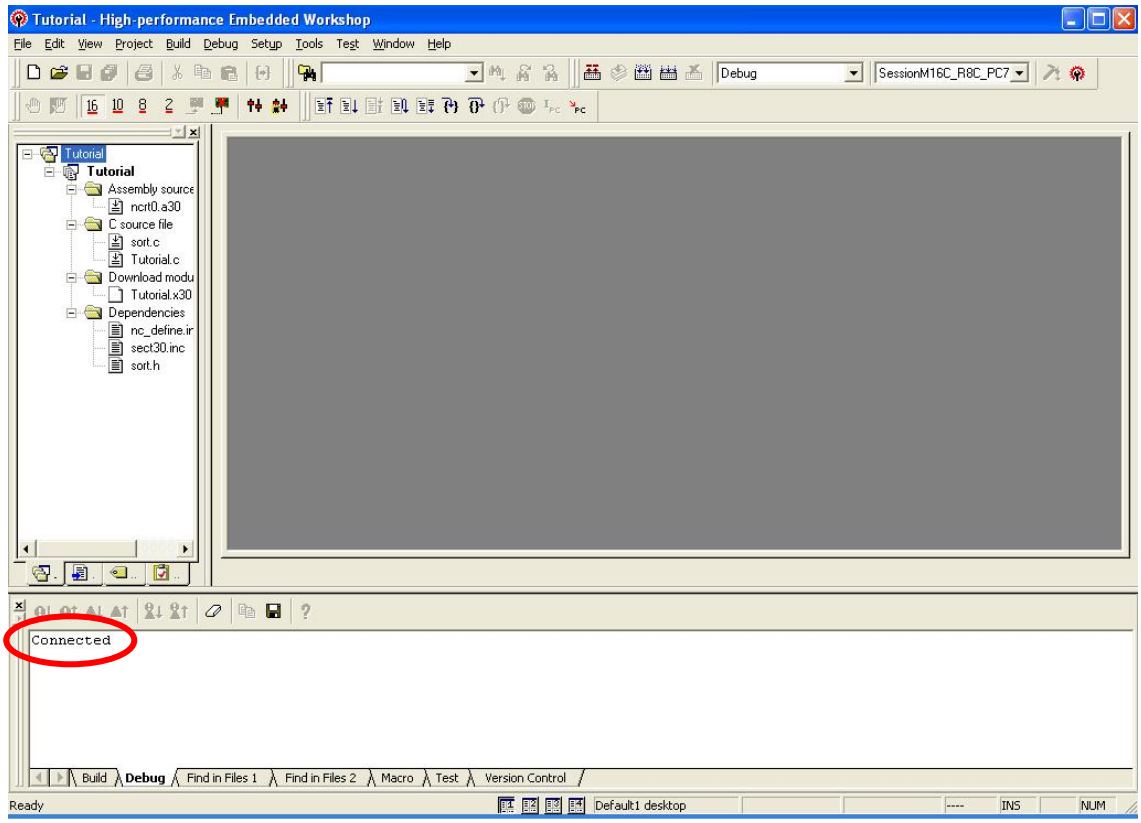


- If your emulator is connected via LPT or LAN instead of USB, visit the Renesas website and see the PC7501 user’s manual.

(6) The MCU Setting dialog box will be displayed. Select “Single-Chip Mode” for Processor Mode on the MCU tab of the dialog box and click OK.



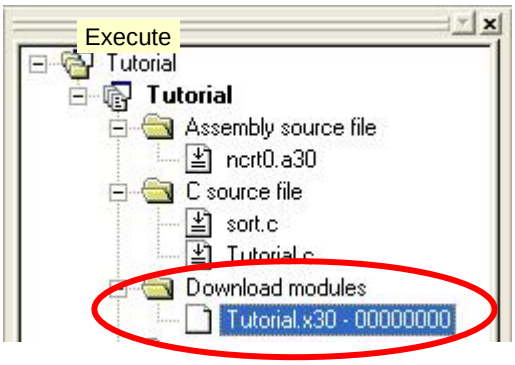
(7) A connection of the PC7501 emulator will be completed, by which the High-performance Embedded Workshop screen becomes ready to operate. When this connection is complete, a message “Connected” is displayed on the Debug tab of the output window.



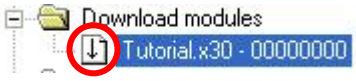
4.3 Downloading the Target Program

First, download the target program.

(1) In the example here, we’ll download the program “Tutorial.x30 - 00000000” appearing beneath Download Modules in Workspace. Double-click it to download.



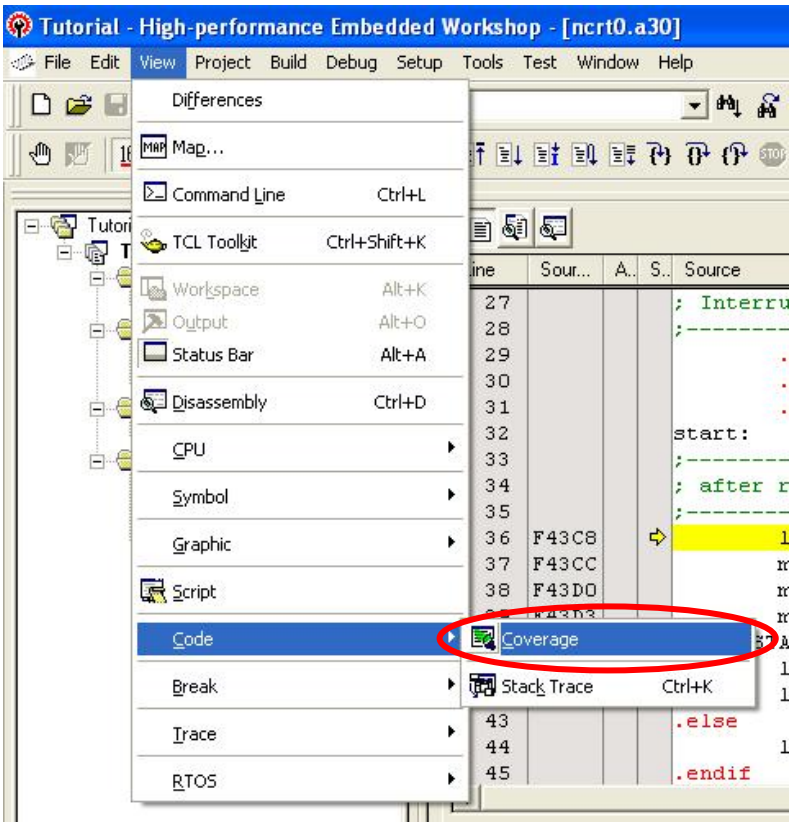
When the program is successfully downloaded, its icon is marked by a down arrow.



4.4 Measuring the C0 Coverage

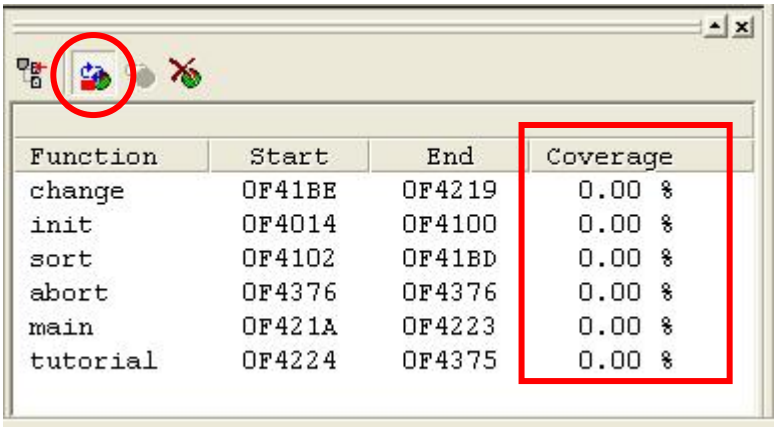
To view the results of coverage measurements, open the coverage window.

(1) From the View menu, choose Code and then Coverage.



(2) The coverage window will open. The window shows the function names of the target program under the heading “Function,” as well as the start addresses under “Start,” the end addresses under “End” and the C0 coverage measurement result under “Coverage.” At this point in time, the values under “Coverage” are 0.00% because the program is not run yet.

In the explanation in this document, the Auto Display Update icon is “enabled” (the icon is depressed when enabled).

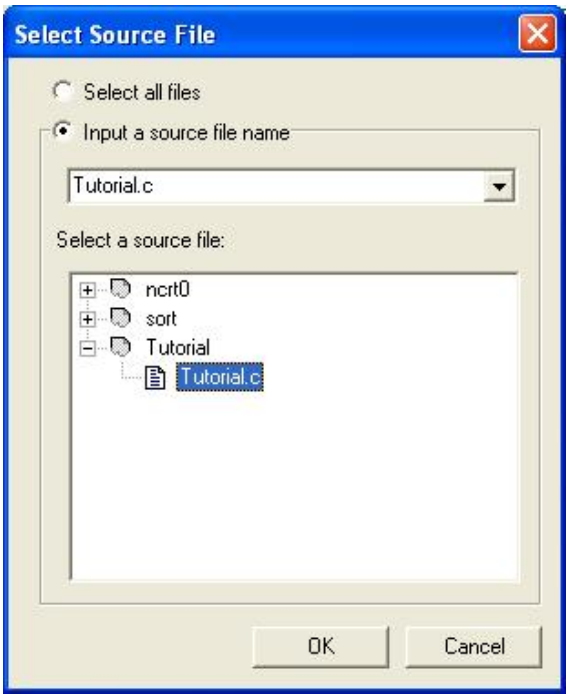


Function	Start	End	Coverage
change	0F41BE	0F4219	0.00 %
init	0F4014	0F4100	0.00 %
sort	0F4102	0F41BD	0.00 %
abort	0F4376	0F4376	0.00 %
main	0F421A	0F4223	0.00 %
tutorial	0F4224	0F4375	0.00 %

• Description of the icons

Source selection: Selects the source file for which the results of coverage measurements are displayed.

When you select the Source Select icon, the Select Source Files dialog box is displayed, allowing you to display only a selected source file in the coverage window.



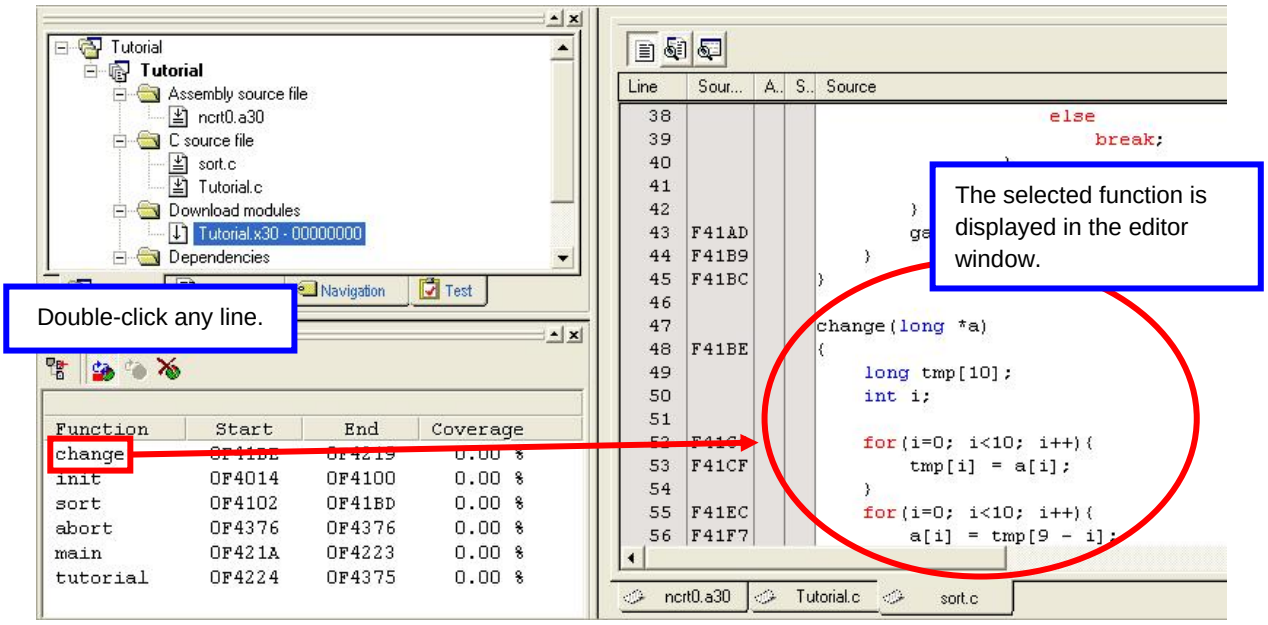
[the icon is depressed when enabled]

Automatic display update: Automatically updates the displayed coverage measurement result when the target program stops.

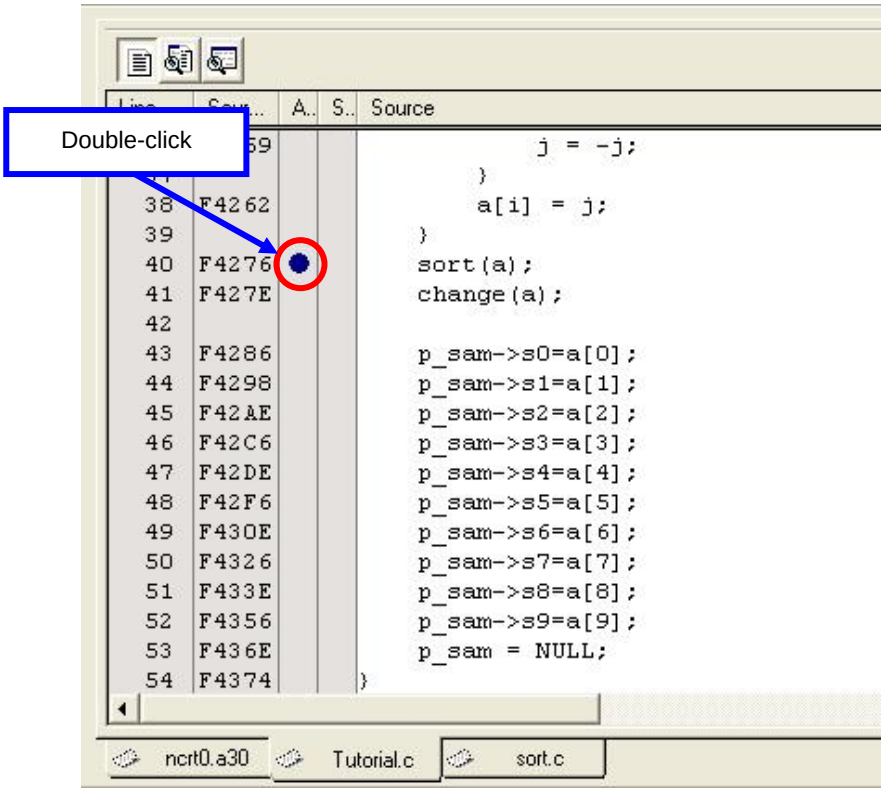
Display update: Updates the displayed coverage measurement result.

Initialization: Initializes the coverage measurement result.

(3) Double-click any function line displayed in the coverage window. The function you've selected will be displayed in the editor window.

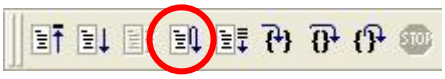


(4) We'll now try stopping the program immediately before a call to the sort function to see how the C0 coverage will be measured.
Double-click in the Address Match Breakpoint column at line No. 40 of the source file "Tutorial.c."

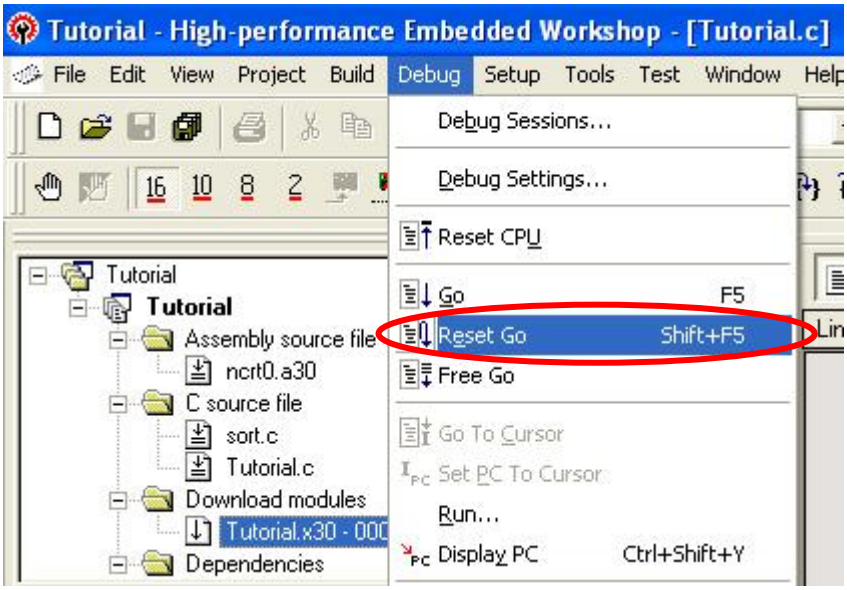


Since we've set the program to break immediately before the source address "F4276" of the sort function is executed, the functions change and abort present ahead of it are not executed.

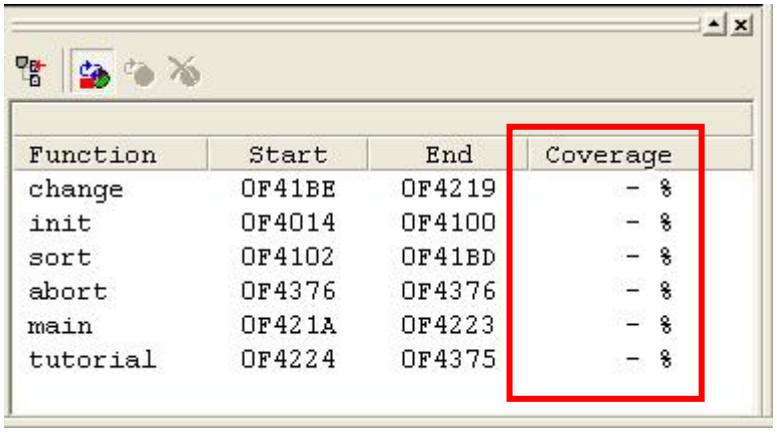
(5) Run the target program to measure the C0 coverage. Select the Run After Reset icon to run the program.



The program can also be run from the Debug menu.

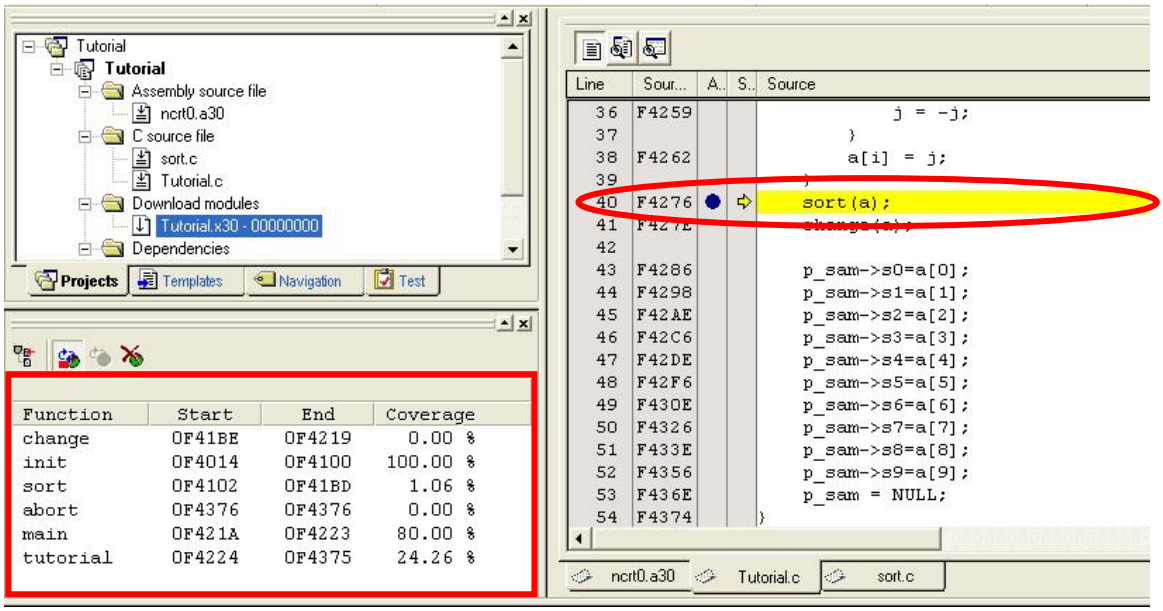


(6) While a C0 coverage measurement is in progress, the “Coverage” values in the coverage window are shown as “– %.”



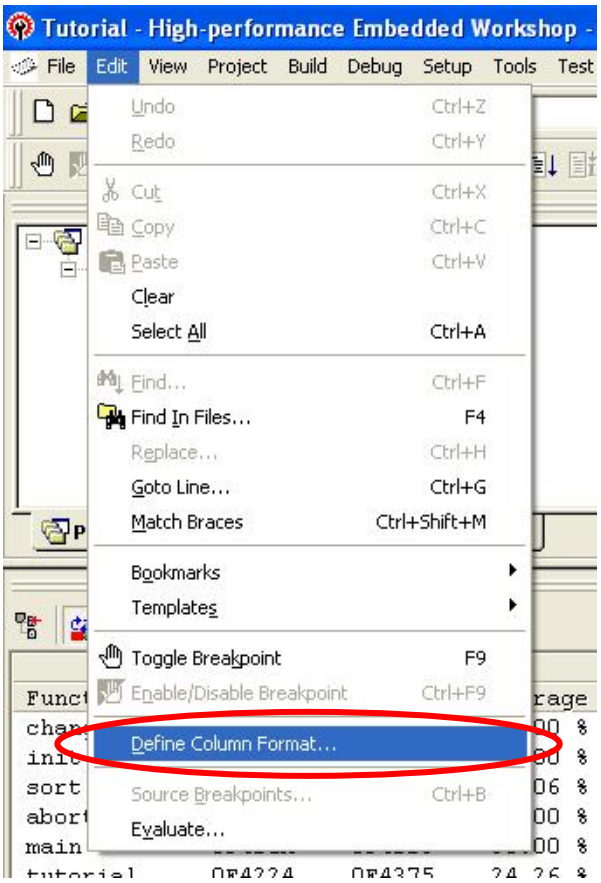
Function	Start	End	Coverage
change	0F41BE	0F4219	– %
init	0F4014	0F4100	– %
sort	0F4102	0F41BD	– %
abort	0F4376	0F4376	– %
main	0F421A	0F4223	– %
tutorial	0F4224	0F4375	– %

(7) The program will stop at the breakpoint, showing the result of the C0 coverage measurement in the coverage window.

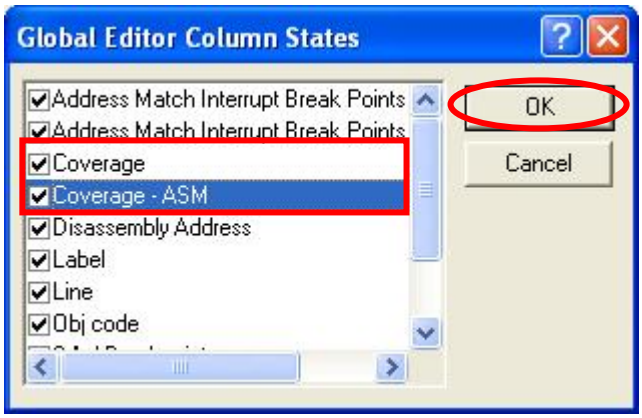


In this case, the function change to be called after the function sort and the function abort used for interrupts were not executed. Therefore, as you can see, their measured values are shown as “0.00%.”

(8) The result of C0 coverage can also be verified in the editor window. Choose “Set Display Columns” from the Edit menu.



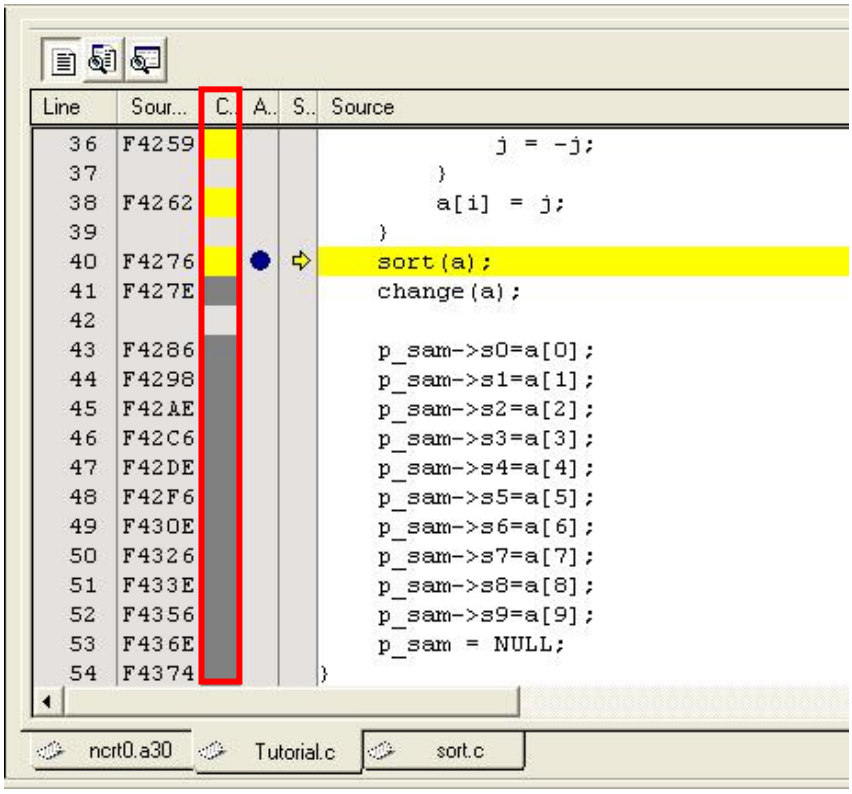
(9) The Entire Column State of Editor dialog box will be displayed. Select the “Coverage” and “Coverage-ASM” check boxes in the dialog box and click OK.



Coverage: Permits you to verify C0 coverage in C source units.

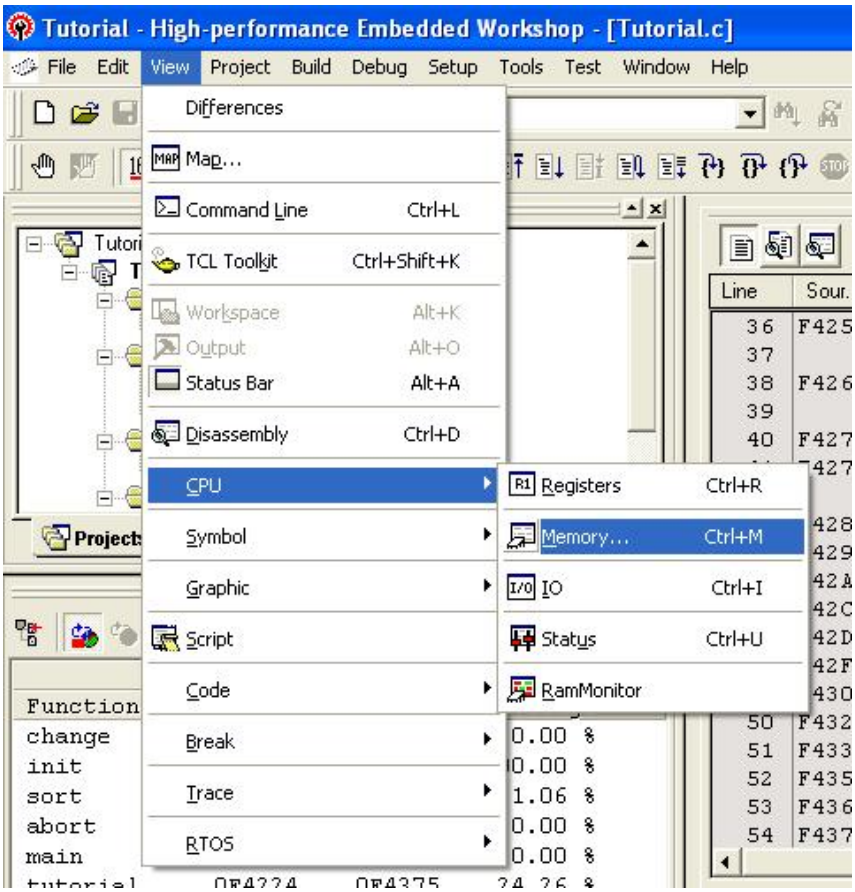
Coverage-ASM: Permits you to verify C0 coverage in assembly units.

(10) The “Coverage” column will be added in the editor window, allowing you to verify C0 coverage visually. The yellow part of this column represents executed instructions, and the gray part represents unexecuted instructions.



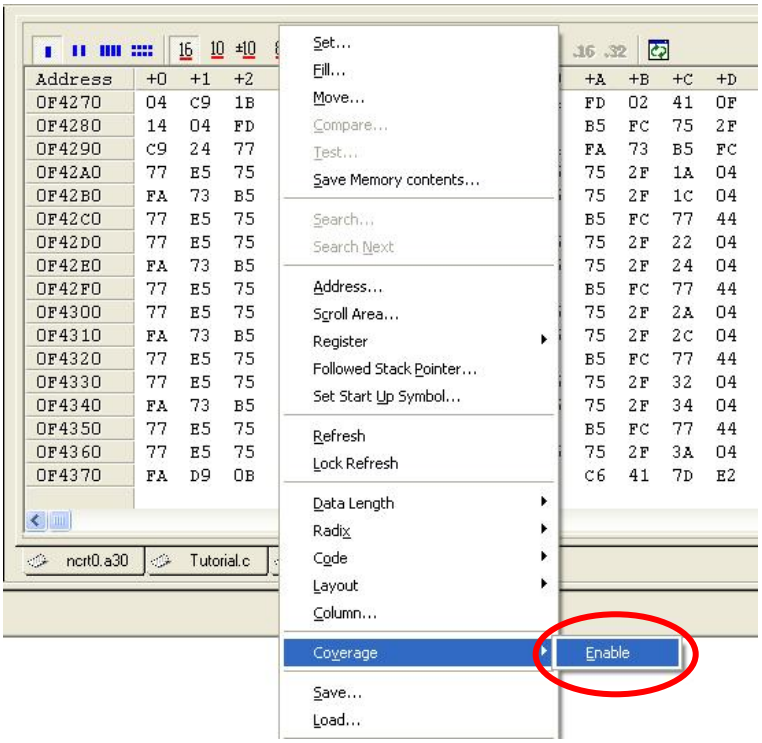
In mixed mode or disassembly mode too, it is possible to verify C0 coverage visually in the same way.

(11) What's more, it is possible to verify C0 coverage visually in address units using the memory window. Choose CPU and then Memory from the View menu.

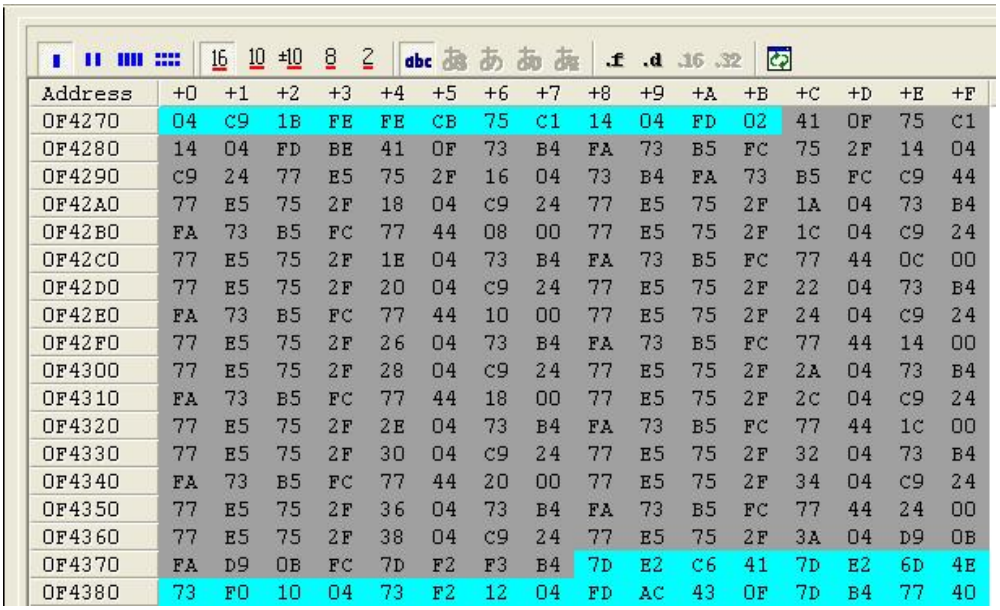


(12) The Display Start Address dialog box will be displayed. Set "H'F4270" for Display Start Address in this dialog box and click OK.

(13) The memory window will open. Right-click in the window and from the ensuing menu, choose Coverage and then Enable.

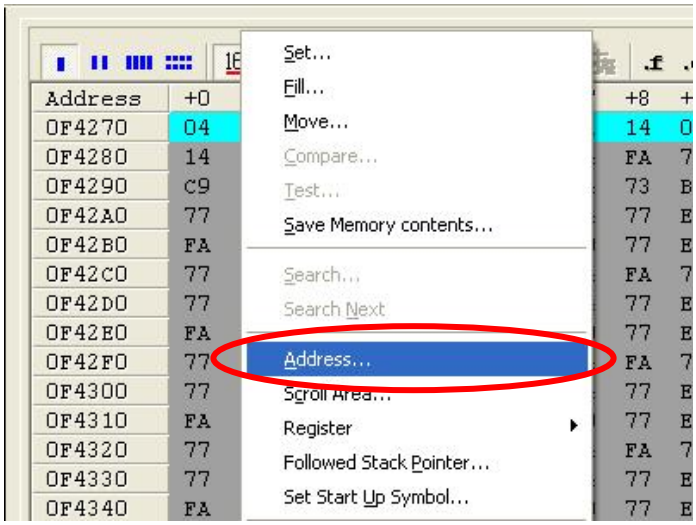


(14) The memory window will be color-coded. Shown in the blue part are the executed instructions, and those in the gray part are the unexecuted instructions.

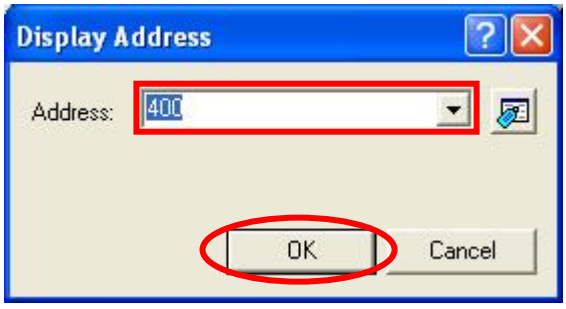


Address	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0F4270	04	C9	1B	FE	FE	CB	75	C1	14	04	FD	02	41	0F	75	C1
0F4280	14	04	FD	BE	41	0F	73	B4	FA	73	B5	FC	75	2F	14	04
0F4290	C9	24	77	E5	75	2F	16	04	73	B4	FA	73	B5	FC	C9	44
0F42A0	77	E5	75	2F	18	04	C9	24	77	E5	75	2F	1A	04	73	B4
0F42B0	FA	73	B5	FC	77	44	08	00	77	E5	75	2F	1C	04	C9	24
0F42C0	77	E5	75	2F	1E	04	73	B4	FA	73	B5	FC	77	44	0C	00
0F42D0	77	E5	75	2F	20	04	C9	24	77	E5	75	2F	22	04	73	B4
0F42E0	FA	73	B5	FC	77	44	10	00	77	E5	75	2F	24	04	C9	24
0F42F0	77	E5	75	2F	26	04	73	B4	FA	73	B5	FC	77	44	14	00
0F4300	77	E5	75	2F	28	04	C9	24	77	E5	75	2F	2A	04	73	B4
0F4310	FA	73	B5	FC	77	44	18	00	77	E5	75	2F	2C	04	C9	24
0F4320	77	E5	75	2F	2E	04	73	B4	FA	73	B5	FC	77	44	1C	00
0F4330	77	E5	75	2F	30	04	C9	24	77	E5	75	2F	32	04	73	B4
0F4340	FA	73	B5	FC	77	44	20	00	77	E5	75	2F	34	04	C9	24
0F4350	77	E5	75	2F	36	04	73	B4	FA	73	B5	FC	77	44	24	00
0F4360	77	E5	75	2F	38	04	C9	24	77	E5	75	2F	3A	04	D9	0B
0F4370	FA	D9	0B	FC	7D	F2	F3	B4	7D	E2	C6	41	7D	E2	6D	4E
0F4380	73	F0	10	04	73	F2	12	04	FD	AC	43	0F	7D	B4	77	40

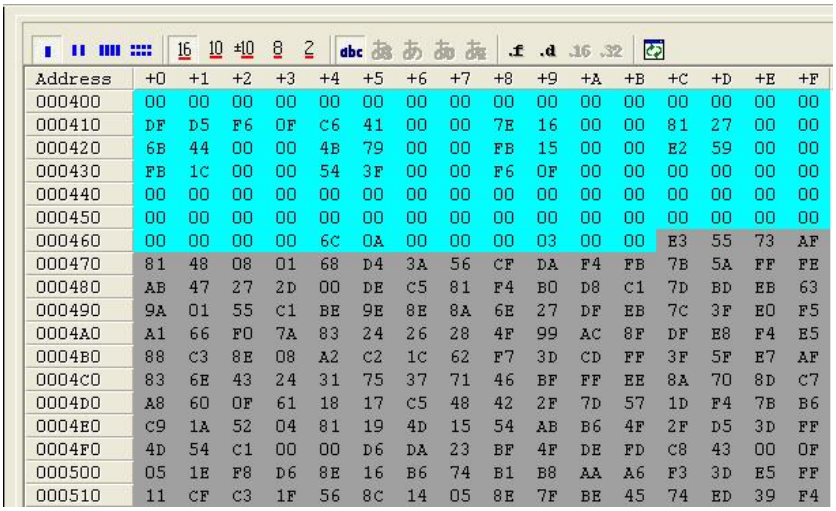
(15) Try verifying the content of the internal RAM following the same procedure. Right-click in the memory window and from the ensuing menu, choose Start Address. The PC7501's coverage facility permits you to measure the coverage of data areas (whether addresses accessed), not just the coverage of instruction execution.



(16) The Display Start Address dialog box will be displayed. Set "H'400" for Address in this dialog box and click OK.



(17) The window permits you to verify the variables in the internal RAM that were accessed and those that were not accessed.

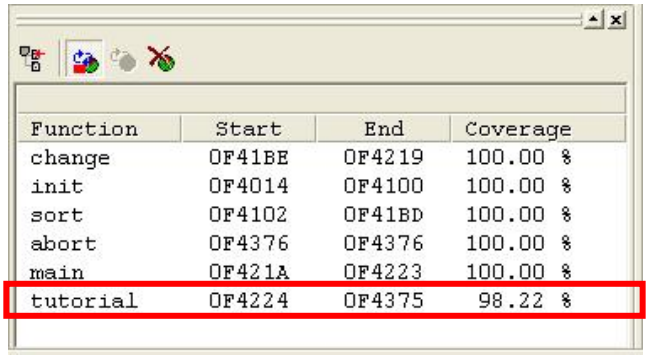


These coverage facilities allow you not only to verify the validity of a test program, but also measure the usage rate, etc. of the RAM space in actual program operation.

5. Frequently Asked Questions

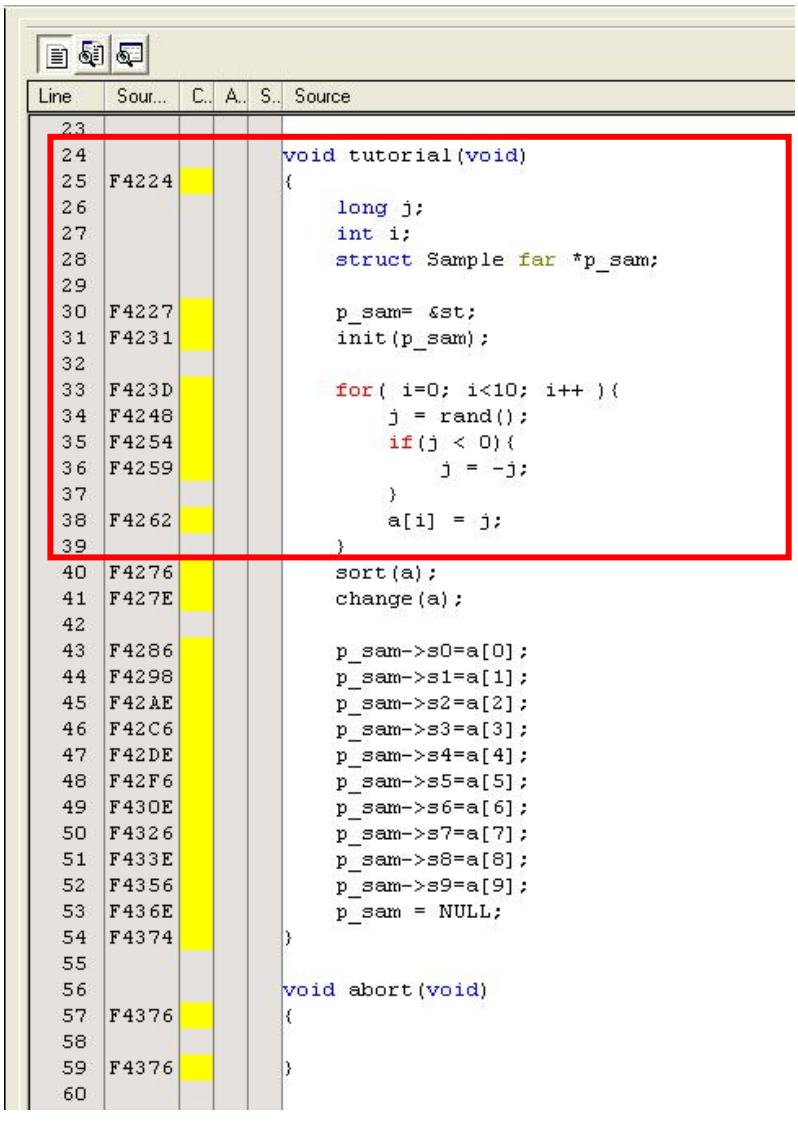
5.1 Why do the measurement result in source units and that in the coverage window differ?

Run the sample program once up until it ends and see the result of C0 coverage measurement. It will look like the one shown below.



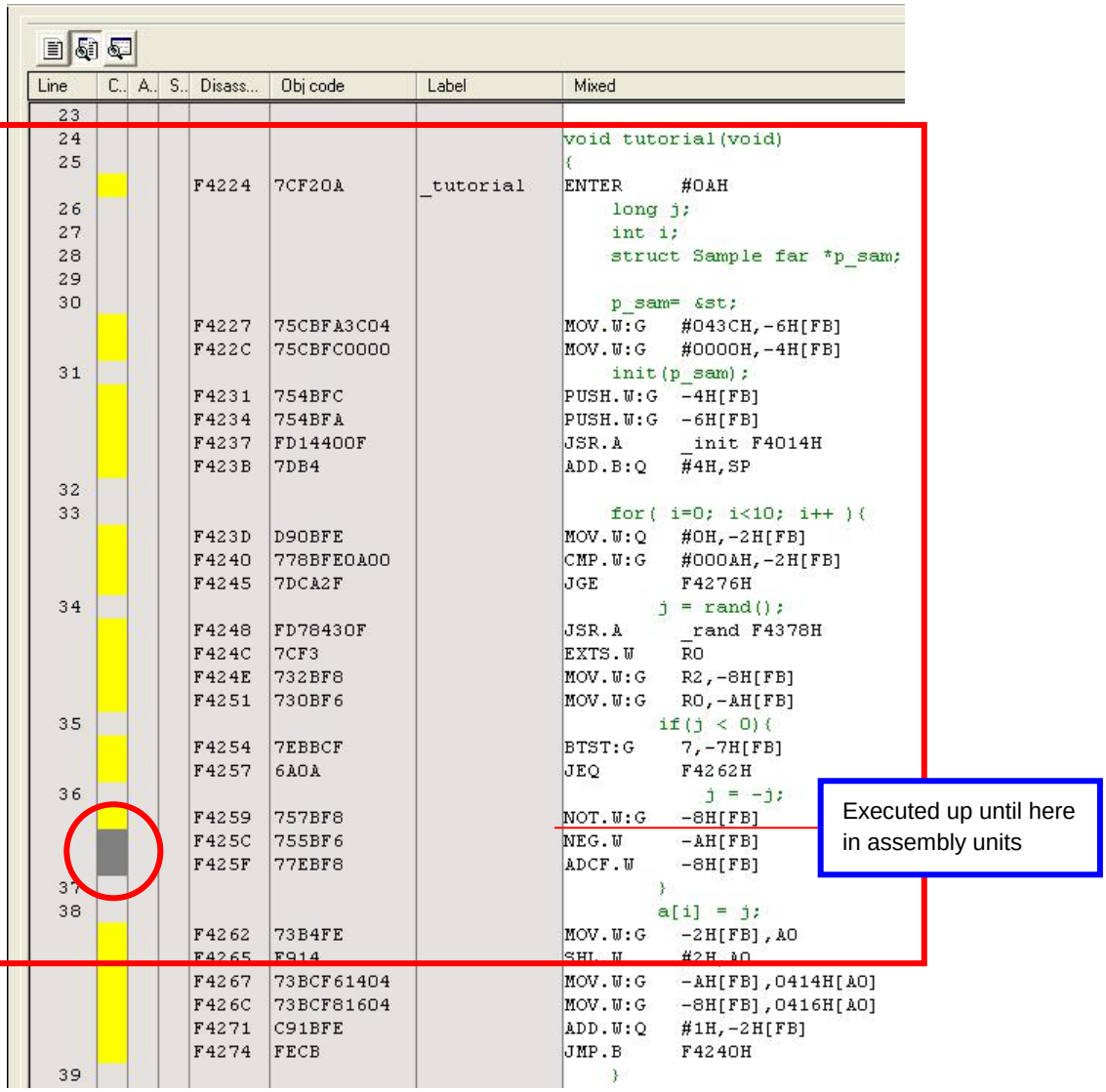
Function	Start	End	Coverage
change	0F41BE	0F4219	100.00 %
init	0F4014	0F4100	100.00 %
sort	0F4102	0F41BD	100.00 %
abort	0F4376	0F4376	100.00 %
main	0F421A	0F4223	100.00 %
tutorial	0F4224	0F4375	98.22 %

Taking a look at the C0 coverage of the function tutorial in C source units, you'll find that the function appears to have been executed in whole.



Line	Sour...	C..	A..	S..	Source
23					
24					void tutorial(void)
25	F4224				{
26					long j;
27					int i;
28					struct Sample far *p_sam;
29					
30	F4227				p_sam= &st;
31	F4231				init(p_sam);
32					
33	F423D				for(i=0; i<10; i++){
34	F4248				j = rand();
35	F4254				if(j < 0){
36	F4259				j = -j;
37					}
38	F4262				a[i] = j;
39					}
40	F4276				sort(a);
41	F427E				change(a);
42					
43	F4286				p_sam->s0=a[0];
44	F4298				p_sam->s1=a[1];
45	F42AE				p_sam->s2=a[2];
46	F42C6				p_sam->s3=a[3];
47	F42DE				p_sam->s4=a[4];
48	F42F6				p_sam->s5=a[5];
49	F430E				p_sam->s6=a[6];
50	F4326				p_sam->s7=a[7];
51	F433E				p_sam->s8=a[8];
52	F4356				p_sam->s9=a[9];
53	F436E				p_sam = NULL;
54	F4374				}
55					
56					void abort(void)
57	F4376				{
58					
59	F4376				}
60					

However, if the C0 coverage of the function tutorial is verified in assembly units, you'll find that part of the function was unexecuted.



The reason why such a phenomenon occurs is because a prefetch is involved. Instructions are loaded into the cache beforehand no matter whether the next data is required. Once prefetched, when in assembly units, instructions are executed after the JEQ instruction up until the next instruction. When this is verified in C source units, instructions are assumed to have been executed up until "j = –j," so that it appears on the surface that all instructions were executed.

If you want to verify C0 coverage at the source level, we recommend verifying the result in mixed mode or disassembly mode.

6. Related Documents

The PC7501 emulator and the HEW have numerous other convenient facilities and features not discussed in this document. Detailed specifications, technical information, limitations and other useful information on each product are described in the related documents listed below. Please see these manuals along with this document.

[PC7501 emulator related documents]

- M16C R8C PC7501 Emulator Debugger User's Manual (for the M16C/60, M16C/30, M16C/Tiny and R8C/Tiny series)
- M16C R8C PC7501 Emulator Debugger Release Notes (for the M16C/60, M16C/30, M16C/Tiny and R8C/Tiny series)
- PC7501 Setup Guide (M16C Family Emulator)
- PC7501 User's Manual (M16C Family Emulator)

[High-performance Embedded Workshop related documents]

- High-performance Embedded Workshop User's Manual
- High-performance Embedded Workshop Release Notes

[CPU related documents]

- M16C/60 Series Hardware Manual
- M16C/60, M16C/20, M16C/Tiny Series Software Manual

[M16C/60 series C compiler related documents]

- M3T-NC30WA C Compiler Guidebook (C compiler package for the R8C/Tiny, M16C/60, M16C/30, M16C/20, M16C/10 and M16C/Tiny series)
- M3T-NC30WA Assembler User's Manual (C compiler package for the R8C/Tiny, M16C/60, M16C/30, M16C/20, M16C/10 and M16C/Tiny series)

To see more information on the PC7501 emulator, please visit the Renesas websites given below.

Japan site: http://japan.renesas.com/pc7501

Global site: http://www.renesas.com/pc7501

Home Page and Where to Contact for Support

Renesas Technology home page
http://www.renesas.com/

Where to contact
http://www.renesas.com/inquiry

Revision Record

Rev.	Issue date	Contents of revision	
		Page	Points
1.00	2009.2.13	—	First edition issued

Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (http://www.renesas.com)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
   (1) artificial life support devices or systems
   (2) surgical implantations
   (3) healthcare intervention (e.g., excision, administration of medication, etc.)
   (4) any other purposes that pose a direct threat to human life
   Renesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.

© 2009. Renesas Technology Corp., All rights reserved.